Liver Disease Prediction

Ali Saeidi Ashtiyani, Ragini Dwivedi and Akshay kumar Gyara

Department of Computer Engineering, Data Mining (CMPE255)

Email: ali.saeidiashtiyani@sjsu.edu, ragini.dwivedi@sjsu.edu, and akshaykumar.gyara@sjsu.edu,
https://github.com/alisaeidi92/Liver-Disease-Prediction

Abstract

Numerous patients suffer every year of liver disease. However, without invasive procedures like surgery, it's impossible to verify the presence of disease with certainty. According to Mayo Clinic, "Abnormal liver function test results don't always indicate liver disease." As a result, the inability for medical professionals to deliver confident answers leaves patients with fear, and doctors with an uncertain pathway to treatment. Our goal is to resolve these problems by increasing the accuracy of current tests using data mining and machine learning strategies.

Introduction

As Machine Learning (ML) is getting more advanced and heavily researched, its applications in health and medicines is becoming more popular. Data mining and machine learning can be used to diagnose patients with certain diseases based on their test results.

This project focuses on detecting patients with liver disease based on their Comprehensive Metabolic Panel (CMP) blood work results. We have processed the data and developed multiple ML algorithms that can classify sick patients based on their bloodwork with high accuracy. Moreover, this project provides a great insight and comparison between different methods of data mining and pros and cons of different ML algorithms for medical data.

Backgrounds and Similar Works

There have been countless number of researches and projects on medical datasets, specifically for liver diseases. This project is based on a dataset found on kaggle website named as Liver Disease Lab Data[1]. There have been no previous work or tasks on this specific dataset, however, other datasets such as Indian Liver Patient Records[2] have been heavily researched and multiple solutions were provided. Some works suggested working with CNN algorithms that provide 100% accuracy while some other methods demonstrated a good accuracy using ML models such as SVM or KNN. There have been other works with lower accuracy with Logistic Regression and Naïve Bayes algorithms.

Data and Features

The dataset has 10 columns as features and 1 column as the target class. Features are namely; Age, Total Bilirubin, Direct Bilirubin, Alkaline Phosphotase, Alamine Aminotransferase, Aspartate Aminotransferase, Total Proteins, Albumin and Albumin and Globulin_Ratio which are taken from CMP of each patient. The target class named Liver Disease indicates the presence of liver disease for a patient. Figure 1 shows the unbalanced target class in this dataset.

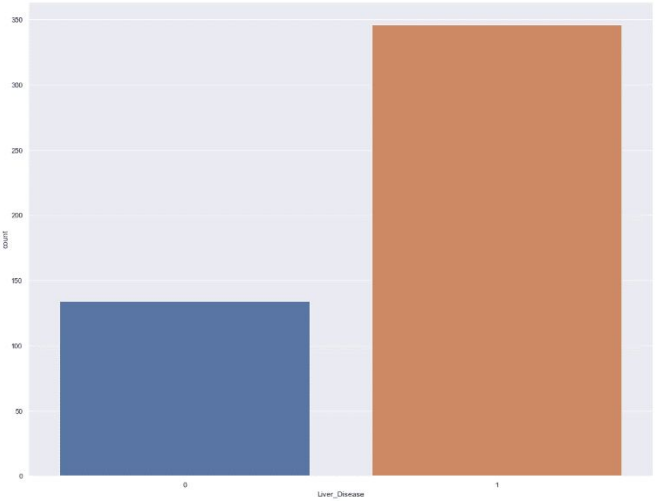

Figure 1- Unbalanced Liver Disease Instances

The imbalance in the dataset will definitely have negative effects in learning of ML algorithms from the dataset. Also,

Figure 2 shows the imbalance in the Gender and the fact that men are more likely to be sick than women.

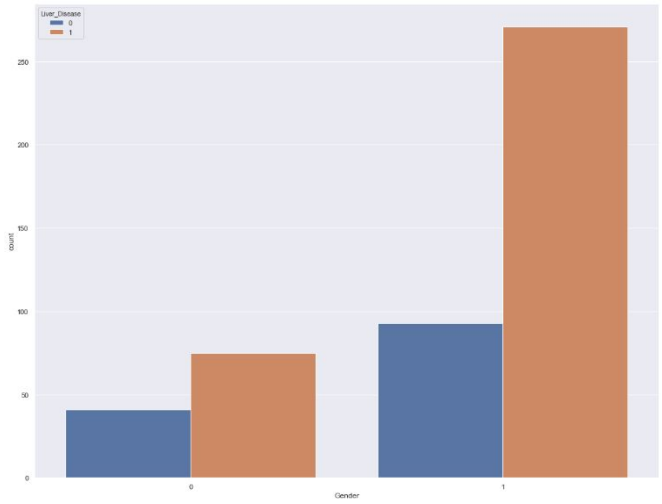

Figure 2- Imbalance between Male and Female

Figure 3 shows the relationship between aging and Alamine Aminotransferase levels and liver disease.

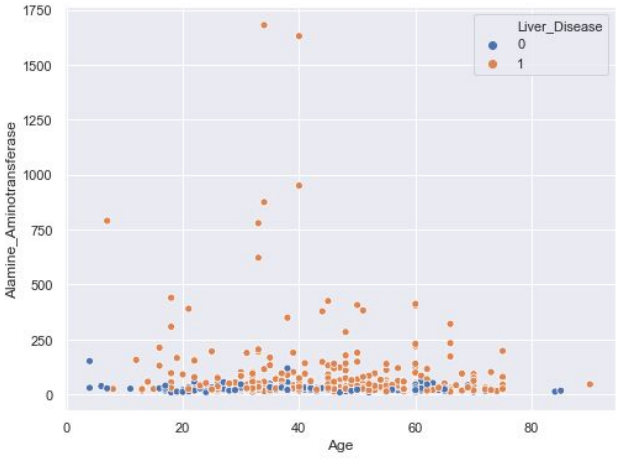

Figure 3- Alamine Aminotransferase vs Aging

As we can see, there's a barely a relationship between Alamine Aminotransferase levels and aging. However, we can conclude that under the two conditions of being older or having high levels of Alamine Aminotransferase patients are at higher risk of having liver disease.

Figure 4 shows the relationship between Alamine Aminotransferase and Albumin.

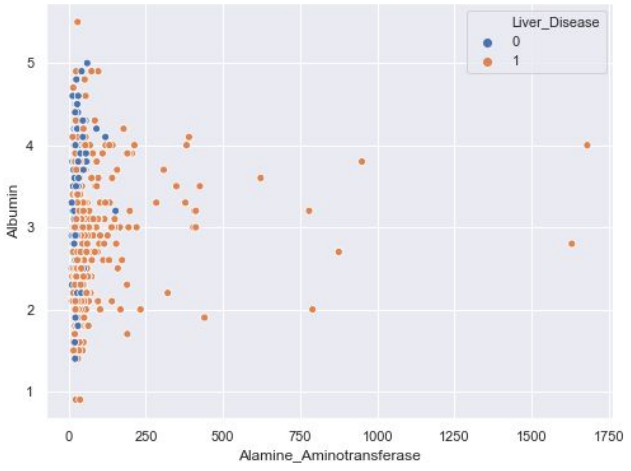

Figure 4- Alamine Aminotransferase vs Albumin

As we can see, there is no obvious relationship between Alamine Aminotransferase and Albumin levels, however we can conclude that lower levels of Albumin and high levels of Alamine Aminotransferase patients are at higher risk of liver disease.

Processing the Data

In this project a good number of pre-processing methods have been developed and used for some algorithms. Since different ML algorithms were used in this project, different processing had to be done to tune the parameters and reach the ideal accuracy. The categorical data such as gender had to be converted to binary format. As the number of rows with missing value were very small, they were removed in this project.

As we can see in Figure 3, some features in our dataset are highly correlated. Such correlation in data may create problems if the dataset is not large enough. Since the dataset used in this project contains only a few hundred instances, the high correlation between columns should be handled. The best method to overcome such issues is to use PCA which is a method for dimensionality conversion and reduction. Therefore, PCA was used to ensure that each feature is being processed in a different dimension. Also, PCA enables us to take in account the most effective features rather than considering all features equally contributing in the results.

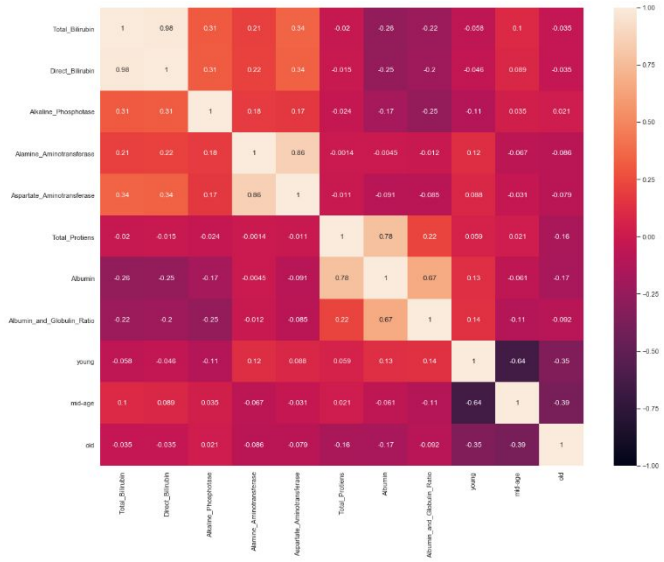

Figure 5- Correlation Between Features

The outliers in this dataset are tricky to deal with. Methods have been developed in the project to deal with outliers in the data however, they were only used in a few of our approaches. The main reason in some features, outliers actually are indications of liver disease. For example, for some liver conditions a certain enzyme may be elevated to 1000s range while the mean is around just 30. Therefore, to deal with outliers in such instances we set the maximum value to be 1000. In this scenario while we partially eliminate the outliers, the information conveyed by such instances are not lost. Figure 4 shows the existence of outliers in this dataset.

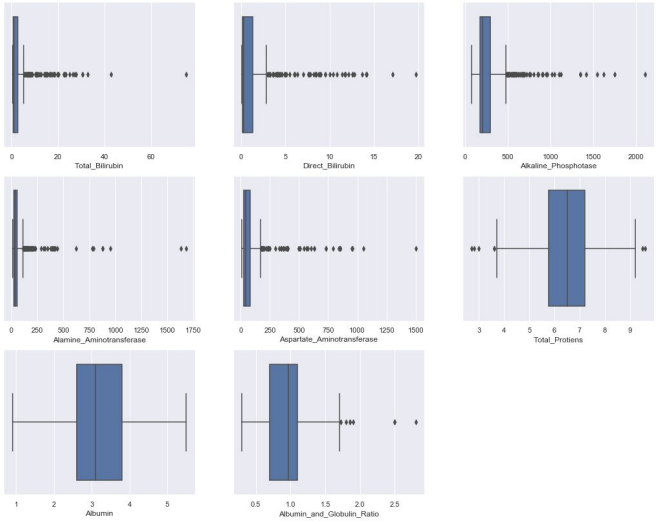

Figure 6- Outliers in Dataset

Algorithms

Random Forest Classifier

Random Forest classifier (RF) is an accurate and a very fast classifier compared to other classifiers. It uses uncorrelated trees as ensembles, each tree providing a classification result and the highest vote will be chosen as the class for that specific instance. Figure 2[3] shows the idea behind RF classifiers.

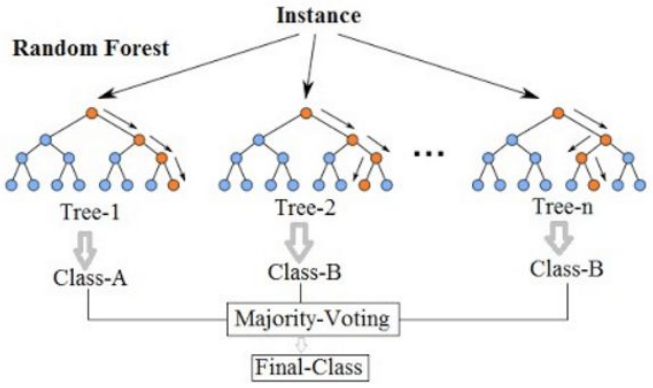

Figure 7- Random Forest Classifier

The RF classifier built in sklearn is used in this project for classification of liver disease. Even though the sklearn RF classifier has numerous parameters to be set, we chose the most important parameters to tune which are namely; n_estimators, criterion, min_samples and max_features. The n_estimators parameter is the number of ensembles or trees used to classify each instance which is the most important attribute in RF classifier. The best n_estimator is either chosen by trial or other methods such as GridSearchCV.

Support Vector Machine

A variety of classification algorithms are used in various medical applications. For unstructured data, classification builds an effectual model for predicting class labels. SVM is a supervised learning computation with related learning algorithms which analyzes data used for regression analysis and classification. SVMs are generally utilized for learning ranking functions, regression or classification. A Support Vector Machine (SVM) isolates the information into two categories of performing grouping and building a N-dimensional hyperplane. SVM depends on statistical learning hypothesis and basic risk minimization rules and has the goal of deciding location of decision boundaries. This is

also known as hyperplanes that produce the optimal separation of classes.

There are multiple training methods for polynomial, multi-layer perceptron classifiers and radial basis functions in which the weight of the network is calculated by solving a quadratic equation problem with linear constraints. There are many kernel functions available to transform the data. One of the most common kernel function are:

1. Linear polynomial
2. Sigmoid
3. Radial basis function

An indicator variable which is considered an attribute and a transformed attribute that is utilized to characterize the hyper plane is known as a feature[18]. Here, picking the most appropriate representation can be taken as feature selection. A lot of features that describe one case is known as a vector. The objective of this modeling is to locate the ideal hyperplane which isolates clusters of vectors in such a manner that cases with one classification of the targetVariable are on one side of the plane and cases with the other classification are on the opposite side of the plane. The vectors closest to the hyperplane are the support vectors[5] as in figure 5[6].

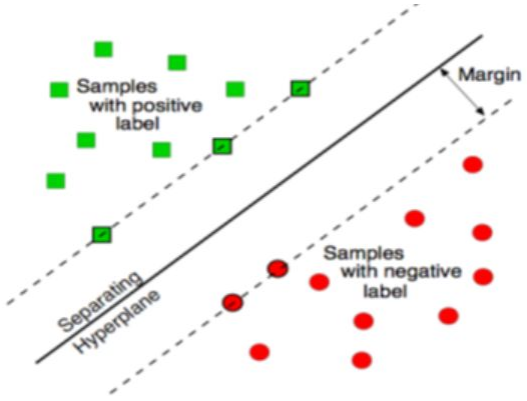

Figure 8- SVM Classifier (Hyperplane)

K-Nearest Neighbors

The K- Nearest Neighbors (KNN) is a supervised learning algorithm that can be used as either a classifier or reggresor. It is a versatile and robust classifier which is commonly used as a benchmark for other complex classifiers such as "Artificial Neural Networks" (ANN) and "Support Vector Machines" (SVM). This classifier has outperformed other classifiers and has been used in various applications such as genetics, pattern recognition, economic forecasting and recommendation systems.

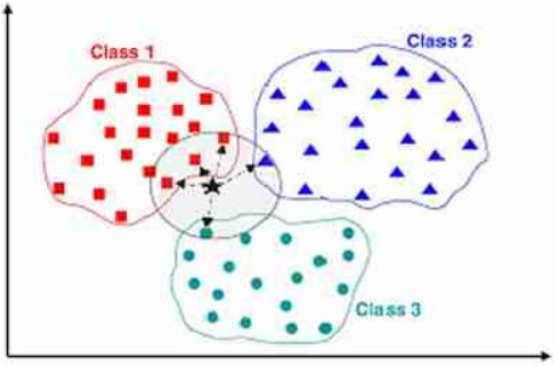

Figure 9- An Example of KNN Classifier

KNN depends on feature similarity i.e. when given a training data set it tries to find the features which are close to classify a given point . In the above figure there are three classes : class 1, class 2 and class 3. It uses the KNN algorithm to estimate or predict the class of black points by finding the nearest neighbours to it. More simply this means given a dataset with observations (X,Y) where X is the predictor and Y is the target or class attribute . Our goal is to find a relation between X and Y and define a function f : X implies to Y i.e. given an unseen observation X, f(X) can accurately predict the output Y.

KNN is also known as Lazy Algorithm . As it doesn't have much parameters to play with. The most important parameter here is "n_neighbors = k". We have to choose the value k in such a way that we can get a best fit possible. Mostly we have to choose k as an odd number as it takes the majority votes of nearest neighbors to predict the class label. Ex if we take k = 6 and we get class 1 as 2, class 2 as 2 and class 3 as 2 , it becomes difficult to decide which class label to be used. The best practical values for k ranges from 5 to 10 [4]. We can find the K value using parameter estimators such as gridsearch or randomly using different k values. Other default parameters are 'weight = uniform' where all the points are equally weighted for each neighborhood , 'metric = minkowski' which is used to measure the similarity .The default distance metric used by KNN is Euclidean distance also known as Minkowski distance.

Results

Random Forest Classifier

To get the best results, We used different combinations of pre-processing methods and parameter tuning. The most important metrics to consider in this part was the recall and f1 scores. Recall score indicates the number of sick patients that were detected from all the sick patients which should be the first priority of this project. We used GridSearchCV to find the best parameters for the model. Moreover, oversampling (over) and PCA were used for some methods. Additionally, two standardization methods were used namely; MinMaxScaler and StandardScaler from sklearn library. Below, we can see the table containing the results of applying combinations of these methods and the results.

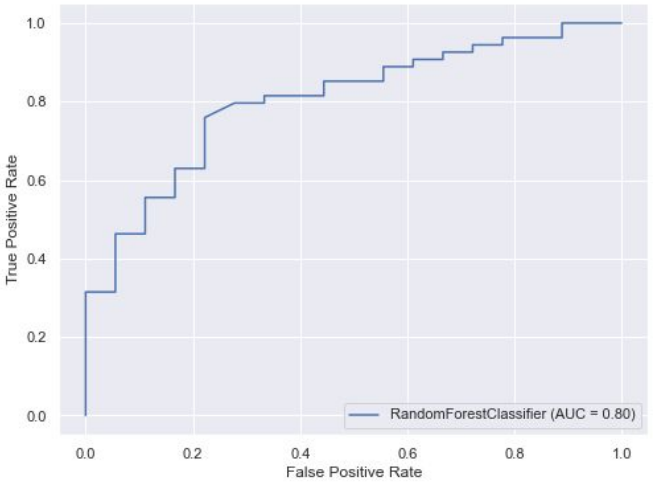

Figure 10- ROC curve of a good performing RF model

PCA	Over #	AgeGrp	outliers	scaler	recal l	F1
No	Yes	Yes	No	MinMax	0.76	0.78
No	Yes	Yes	Yes	MinMax	0.79	0.81
No	No	No	No	MinMax	0.90	0.84
No	Yes	No	No	MinMax	0.75	0.81
No	Yes	No	Yes	MinMax	0.82	0.80
4	No	No	No	MinMax	0.83	0.82
2	No	No	No	MinMax	0.93	0.85
6	No	No	No	MinMax	0.83	0.83
8	No	No	No	MinMax	0.85	0.84
6	Yes	No	No	MinMax	0.72	0.80
No	No	No	No	Std	0.87	0.85
6	No	Yes	Yes	Std	0.83	0.82
6	Yes	Yes	Yes	Std	0.72	0.80

Figure 10 and Figure 11 show the ROC curve and confusion matrix of one of the good performing RF models.

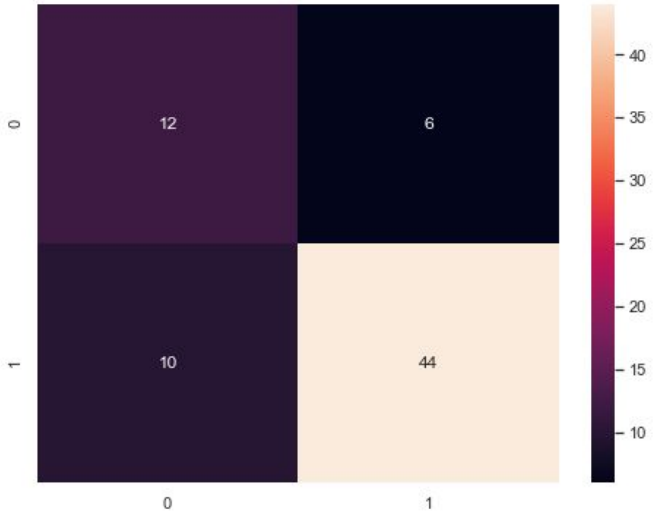

Figure 11- RF Model Confusion Matrix

The recall and F1 score are for the sick patients (target=1). For most methods, there has been a tradeoff between the accuracy on 0 targets and 1 targets meaning when the recall on sick patients is very high, the recall for non-sick patients tends to be too low. Meaning that the model is classifying more patients as sick to be able to capture all the sick patients. However, oversampling and higher PCA value seems to solve this problem to some extent and create a balance between the recall values for sick and un-sick patients.

Support Vector Machine

Before applying SVM, we implemented different preprocessing techniques to fine tune data and remove noise. By looking at the data it is clear that age is the factor for liver disease for both male and female genders. Below figure shows the same:

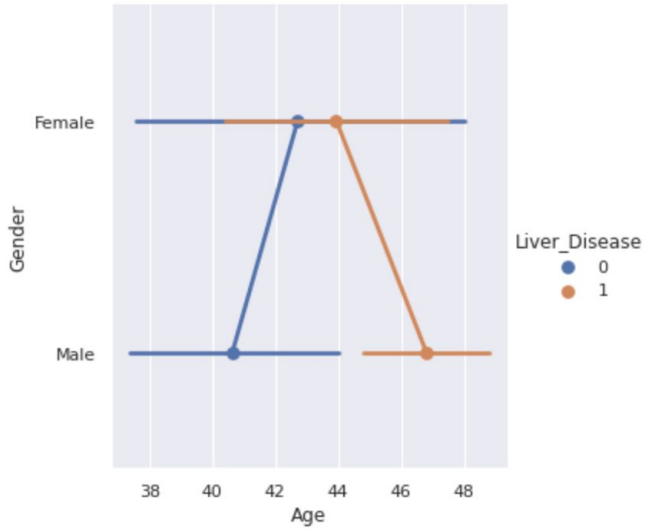

Figure 12- Factor of liver disease base on age

For scaling the data MinMaxScaler was used which translates each feature individually such that it is the given range on the training set. To reduce the dimensionality we used PCA as it helps to improve the distance metric. The n_component value is 6 for calculating PCA. Standardization method was used from sklearn library to preserve zero entries if the feature matrix is sparse. By analyzing the dataset, it was found that different values of features do not have any inherent ordering. To solve this issue One-hot encoding was used. Once the data is ready SVM is applied with a linear kernel, random_state = 9, probability = true and gamma = 0.0000001. Below is the result of SVM application:

	precision	recall	f1-score	support
0	0.52	0.23	0.32	48
1	0.64	0.87	0.74	77
accuracy			0.62	125
macro avg	0.58	0.55	0.53	125
weighted avg	0.60	0.62	0.58	125

Figure 13- Initial result for applying SVM

In the above results, prediction time is 1223 ms and training time is 15239 ms and the accuracy score is 62%.

ROC plot is used to represent TPR(True positive rate) vs FPR(False positive rate). It also represents the performance of the machine learning algorithm. ROC is helpful in our case because simply knowing the number of correct predictions would not be sufficient. Figure 14 and 16 shows the ROC (Receiver Operating Characteristics), accuracy and F-score.

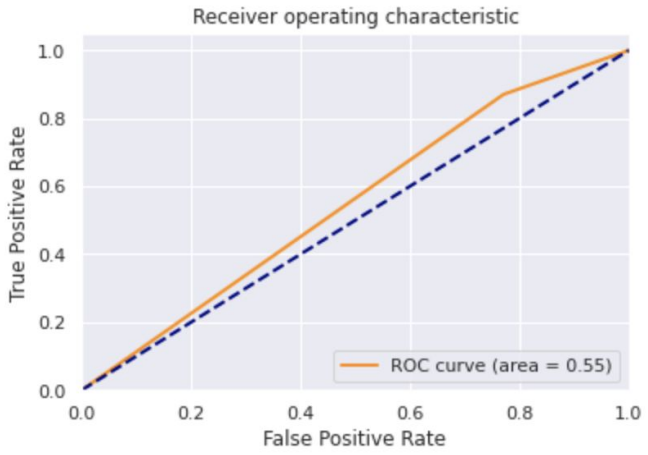

Figure 14- SVM ROC Curve basic model

Once the basic SVM algorithm is applied to the data, the model is fine tuned to make sure the accuracy and results are improved. To improve the performance we have to find the best parameters for SVC using GridSearchCV cross validation. This resulted in choosing the optimised parameters for our model. Optimised value for our model is mentioned below:

```
{'C': 1000, 'gamma': 1e-07}
```

Figure 15- Optimised parameters using GridSearchCV

Once the model was fine tuned, it resulted in increasing the ROC score. The results of new model are mentioned below:

	precision	recall	f1-score	support
0	0.61	0.58	0.60	48
1	0.75	0.77	0.76	77
accuracy			0.70	125
macro avg	0.68	0.67	0.68	125
weighted avg	0.69	0.70	0.69	125

The accuracy of the model increased from 62% to 70%. ROC plot for new model with TPR and FPR is displayed below:

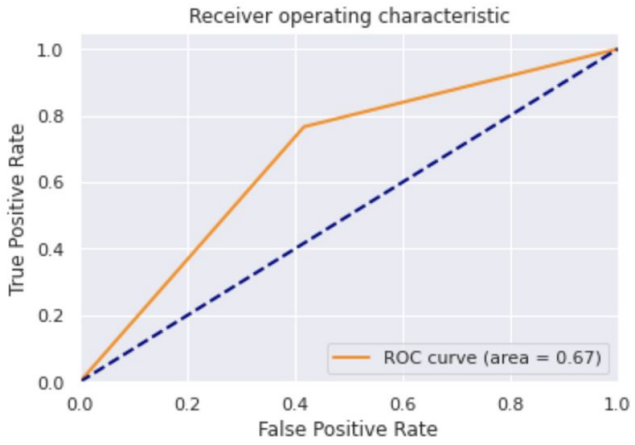

Figure 16- SVM ROC Curve for fine tuned model

According to the data gathered using SVM, it is evident that SVM shows pretty good results in terms of F-beta score which is **0.8131**. However, its AUC(area under the curve) is still low (**0.674784**).

Below figure shows plotting of confusion matrix with normalization and without normalization for the new model

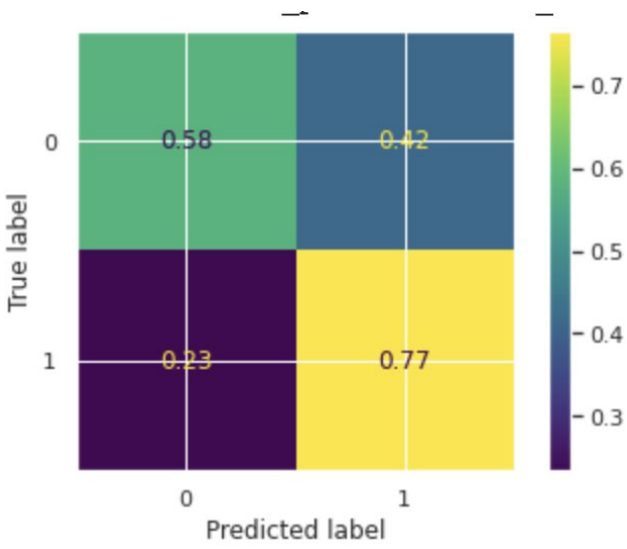

Figure 17- Confusion matrix plot with normalization

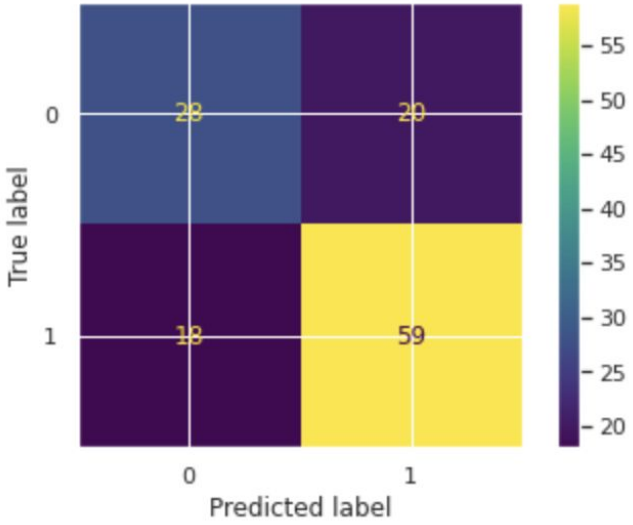

Figure 18- Confusion matrix plot without normalization

K-Nearest Neighbors

For this model initially we just used a basic preprocessing method to remove null value, then changed the categorical variables (gender) to numerical value , created a training and testing set and then trained the model. For this model we just randomly selected values for k. Then tried to reduce the dimensionality using PCA as it helps to improve the distance metric and performed the same procedure as above. Rescaled the data using StandardScalar() and MinMaxScalar() to normalise the data so it can reduce the number of misclassifications. Performed oversampling to balance out the no of '1' and '0' class labels. (1= 347, 0 = 136). Used different distance metrics such as Euclidean distance, Manhattan distance, the hamming distance and cosine distance.. Below is the results table with different approaches and methods where

E = Euclidean Distance
M = Manhattan Distance
H= Hamming Distance
C = Cosine Distance
MM = MinMaxScalar()
SS = StandardScalar()

PCA	Oversample	Distance	K	Scalar	Recall	F1	Accuracy	RMSE
No	Yes	E	1	SS	0.82	0.81	**0.80**	0.421
No	Yes	M	5	SS	0.70	0.75	**0.765**	0.48
No	Yes	H	2	SS	0.69	0.70	**0.822**	0.42
No	Yes	C	5	SS	0.69	0.70	**0.709**	0.53
Yes	Yes	E	3	MM	0.78	0.83	**0.765**	0.484
Yes	Yes	M	5	MM	0.71	0.80	**0.731**	0.484
Yes	Yes	H	1	MM	0.78	0.79	**0.696**	0.557
Yes	Yes	C	3	MM	0.79	0.83	**0.765**	0.491
Yes	No	E	11	SS	0.93	0.85	**0.765**	0.469
Yes	No	M	11	SS	0.91	0.85	**0.772**	0.484
Yes	No	H	2	SS	1.00	0.86	**0.75**	0.49
Yes	No	C	9	SS	0.91	0.86	**0.779**	0.49
No	No	E	19	No	0.98	0.87	**0.775**	0.474
No	No	M	5	No	0.86	0.84	**0.751**	0.496
No	No	H	21	No	0.94	0.86	**0.772**	0.475
No	No	C	16	No	0.94	0.87	**0.7865**	0.465

From the table we can say that the Euclidean distance and Cosine distance had almost the same accuracy and other values for corresponding approaches.We have achieved highest accuracy when we have used oversampled data using SMOTE without doing PCA .As we can see that Recall and precision are not very high though but in this way the model was able to classify class with ‘0’ and ‘1’ very well. And when PCA was used but oversampling wasn't used we got very high Recall values for class‘1’. It is due to imbalance in the no of records i.e. high no of ’1’ (approx 108) and very few no of ‘0’ (approx 37). The model was excellent at predicting ‘1’ but mispredicting class ‘0’. And when No PCA, No scaling was used the accuracy was moderate , The Recall and F1 score are very high for target 1 but bit low for target 0.This is also because we have not used oversampling technique. Finally we can say that we can achieve accuracy and good prediction when we use oversampling . Other factors such as PCA and Scaling didn't have much influence over accuracy or recall.

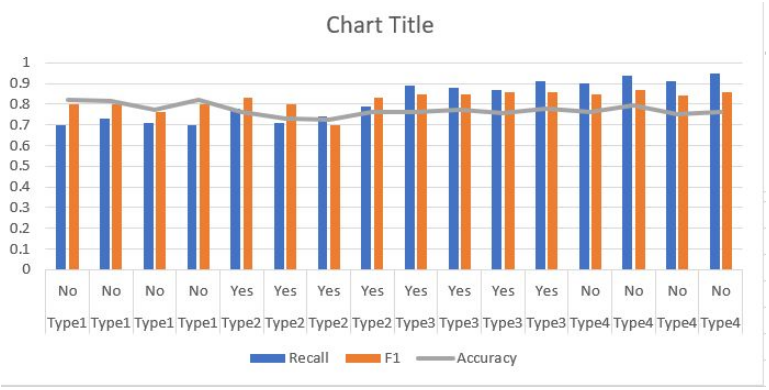

Fig: 18 Analysis on various method used

The Fig 18 is analysis of the data in the above table and shows the trends of Accuracy, Recall and F1 for different techniques.

For selecting the best k value for each model we have used Elbow curve technique. It is a basic step used for any unsupervised learning algorithm to determine the nest value for K. For this we will be using RMSE as the main parameter. RMSE is a measure of difference between predicted value and the observed value. We consider K=1, 2, 3... As K value increases, the RMSE value usually goes down, then becomes stable, and then raises again. Pick the optimal K at the beginning of the stable zone. This is also called the Elbow **Curve Method.**

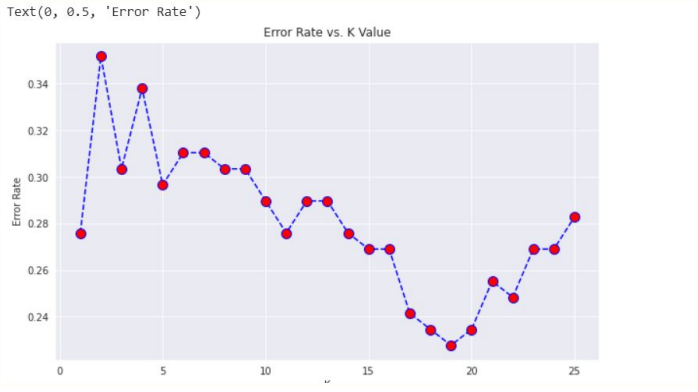

Fig:20 Elbow curve or Error rate

For the graph in fig 20 we can see when K = 19 it stops reducing and from the next value it starts increasing So we can choose K - 19 as the best value.

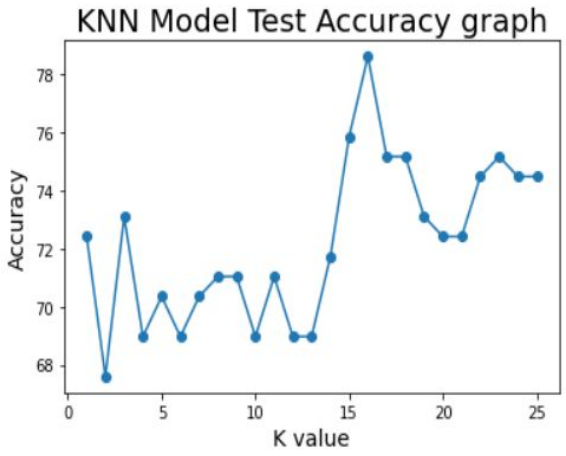

Figure 21- Accuracy graph for KNN with K range 1-25

The above graph is a simple line to represent the changes in accuracy score as the K value changes. we can see that accuracy increased drastically after k =5 and remained high.

A small K value indicates low bias and high variance . As it increases variance decreases and bias increases [4]. As low K value may result in overfitting of data it is better to increase the value of k for better accuracy.

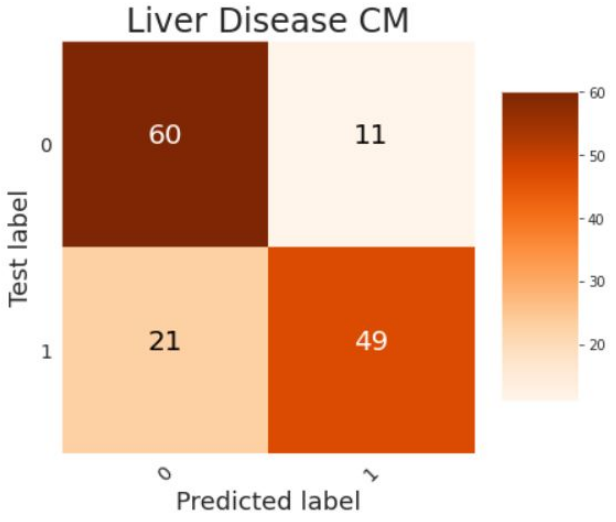

Fig 22 Liver Disease Confusion matrix

We have also plotted a confusion matrix with no. of records classified within each cell. This graph helps us to understand how many records were correctly classified and how many of them were misclassified.

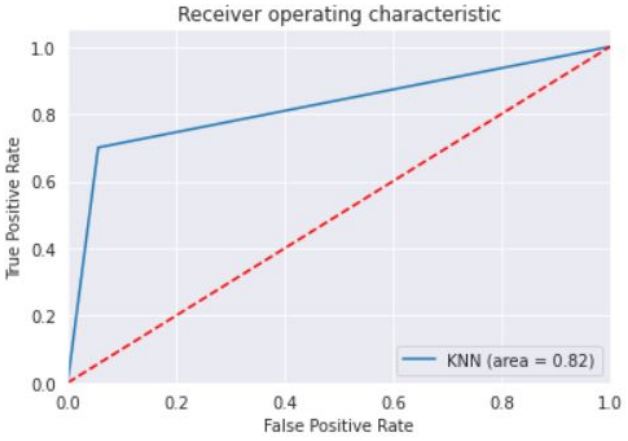

Figure 23- KNN ROC Curve

The above graph is a sample ROC curve. We have used the AreaUnderCover (AUC) method to calculate the ROC curve. We see that ROC curve area is 0.82 which means that the model is very good at predicting class labels. " In general, an AUC of 0.5 suggests no discrimination (i.e., ability to diagnose patients with and without the disease or condition based on the test), 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding " Mandrekar,Jayawant N [7]

Comparison of Models:

For comparing the 3 models Random Forest , SVM and KNN we have chosen a common parameter i.e. ROC curve. ROC curves are frequently used to show in a graphical way the connection/trade-off between clinical sensitivity and specificity for every possible cut-off for a test or a combination of tests

In general, an **AUC** of 0.5 suggests no discrimination (i.e., ability to diagnose patients with and without the disease or condition based on the test), 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding [7].

From fig 10,16 and 23 we can see the best possible ROC curve for Random Forest which is 0.80 , SVM which is 0.67and KNN which is 0.82. By comparing ROC we can say that KNN model was good at classifying the liver disease patients compared to other two models.

Conclusion

The liver disease dataset was relatively small and contained unbalanced data where the number of positive instances were higher than the negative cases. Moreover, the gender distribution in the data was unbalanced. Oversampling was considered in the project in order to handle the imbalance in the dataset.

Other preprocessing methods such as binarizing categorical features or encoding age features into age groups in order to prepare the dataset for machine learning models. Handling outliers was also another approach to try to improve the performance of our models.

The machine learning models used in these projects are Random Forest (RF), Support Vector Machine (SVM) and K-Nearest Neighbours (KNN). As we can see from RF results, the best result was with PCA with 2 components where no other preprocessing was required and MinMax scaler was used. The recall and F1 score for this model was 93% and 85% respectively. On the other hand, the basic SVM model showed a recall and f1 score was 0.87 and 0.8131. The accuracy for the model was 0.62 and ROC score of 0.5496. This accuracy of the model improved when GridSearchCV was used to find optimal parameters and then train a new model. The new model has a recall and f1 score of 0.77 and 0.76. The accuracy of the model went up to 70% and ROC score off 0.6747. Finally, the best performing model was KNN where PCA was applied and n_neighbours were 2 and standard scaler was used. This model performed perfectly at recall of 100% and f1 score of 86%.

Overall, we were able to explore multiple methods of pre-processing and their influence on machine learning models. Multiple machine learning models with various combinations of parameters were used to find the best tune for the dataset. Evidently, the performance of our models were relatively very high and is able to detect patients with liver disease pretty accurately using just the CMP blood results.

Future Work

The liver disease dataset was relatively small. Moreover, it did not contain other types of blood results. In the future, the models used in this project can be trained on larger datasets with more instances and more features from other blood results panels which could help indicating a liver disease. Moreover, a research could be done to create a multiclass problem where our model can classify the type of liver disease each sick patient has which requires a larger dataset with more number of features.

Contributions:

Ali:

- Created a basic logistic regression classifier to start the project with some simple pre-processing methods such as handling missing values and binarizing categorical columns.
- Created and tested multiple random forest models. I used different methods to try to improve the results such as handling outliers, creating age groups and oversampling the minority data.
- Created a GridSearchCV function that can be used for our models to find the best parameters for our data.

Ragini:

- Plotted the factor of liver disease based on age.
- Created SVM hyperplane classifier to further improve the accuracy and F-Score with data pre-processing methods such as data cleaning, removing duplicates and handling null values, standardization using minimax scaler, applied normalization, applying PCA and finding categorical features to build the model.
- Experimented with PCA and did oversampling to improve the accuracy but they did not result in increasing accuracy in case of SVM.
- As a first step of SVM model creation, I have used the baseline predictor for checking our metrics (accuracy, TPR, FPR) on that predictor. After that I used a basic model for prediction.
- Fine tuned the model by using GridSearchCV function to search for optimal parameters for the model. After finding optimal parameters, trained the

model to improve accuracy and performance.

- Created additional metric ROC Curve to check TPR and FPR and the performance of SVM.

Akshay:

- Build a KNN classifier with different parameters , preprocessing and normalisation techniques to achieve good accuracy
- Calculated RMSE which helps to determine the difference between the predicted values to the actual values . Lower the value more efficiently the model is.
- Plotted ROC curve and also calculated AUC .
- Plotted Elbow curve to determine the best value for K.
- Plotted Confusion Matrix to understand the correct and incorrect classification.
- Experimented in different ways using PCA ,Oversampling and scaling . Finally analysed the data from different methods.

References

[1] https://www.kaggle.com/akhan890/liver-disease-lab-data

[2] https://www.kaggle.com/uciml/indian-liver-patient-records

[3] https://towardsdatascience.com/random-forest-classification-and-its-implementation-d5d840dbead0

[4] https://mc.ai/chapter-1-k-nearest-neighbours-classifier/

[5] https://medium.com/machine-learning-researcher/k-nearest-neighbors-in-machine-learning-e794014abd2a

[6] https://medium.com/datadriveninvestor/k-nearest-neighbors-in-python-hyperparameters-tuning-716734bc557f

[7] Surname A and Surname B 2009 *Journal Name* **23** 544

[5] F. Markowetz. "Klassifikationmit support vector Machines". http://lectures.molgen.mpg.de/statistik03/docs/Kapitel16.pdf, 2003.

[6] W. W. Chapman, M.Fizman, B. E. Chapman, and P. J. Haug, "A Comparison of Classification Algorithms to Automatically Identify Chest X-Ray Reports That Support Pneumonia "journal of biomedical informatics,vol.34,pp 4: 14,2001.

[7] https://www.jto.org/article/S1556-0864(15)30604-3/fulltext